



GAMES 003 科研素养课

第三周：技术方案的设计方法



彭思达



高俊

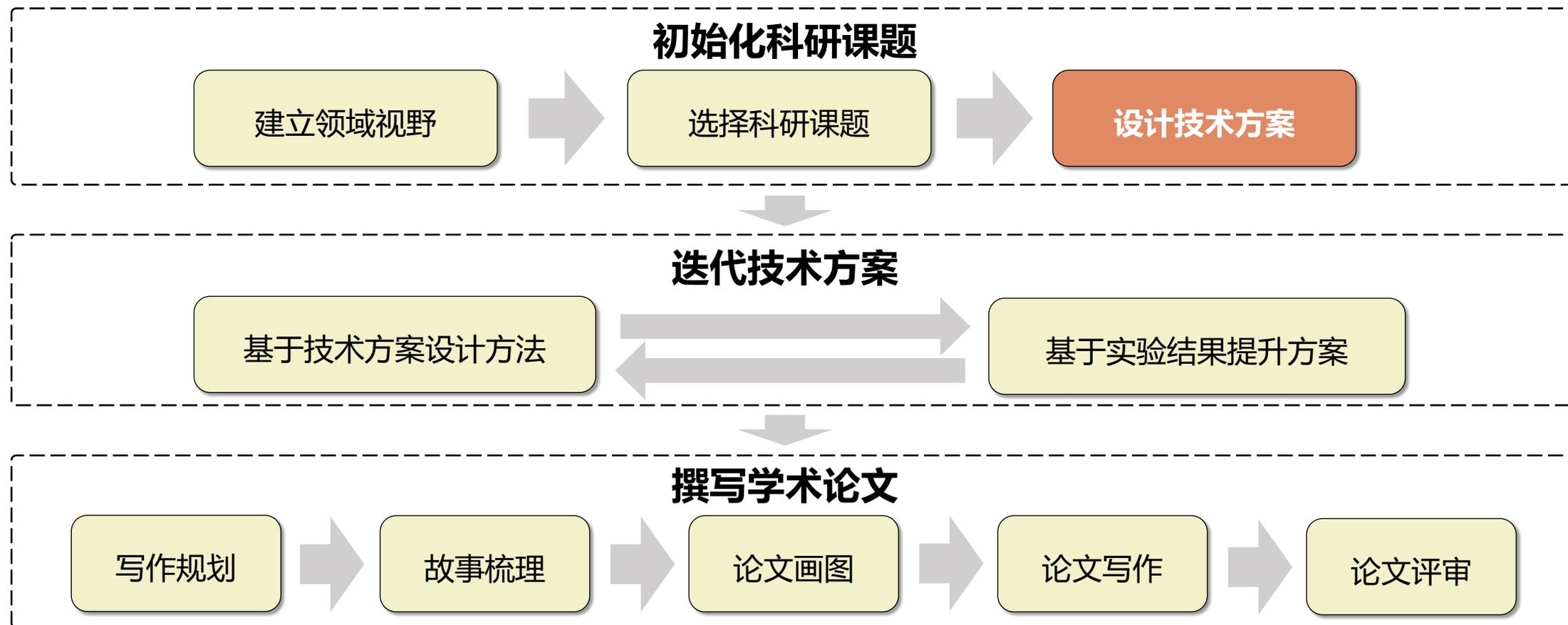


彭崧猷



王倩倩

设计技术方案



方法设计

什么是方法设计

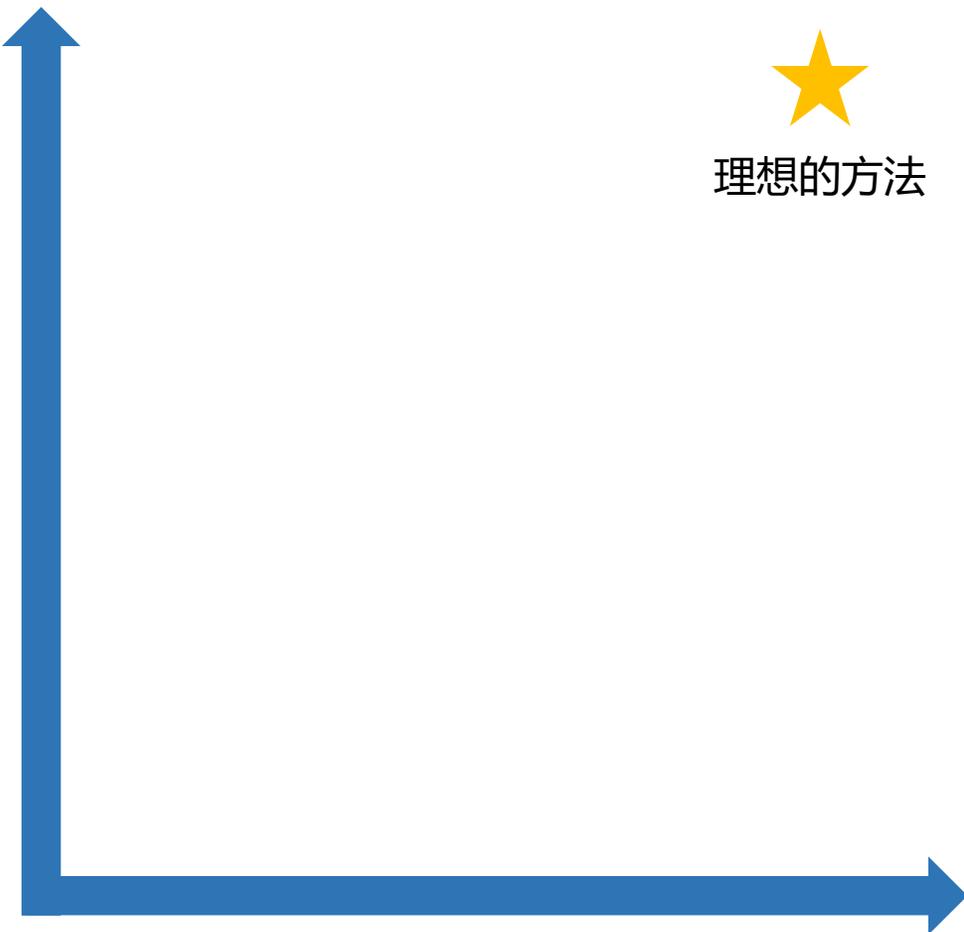
- 在选定课题后，我们需要设计一种新颖的pipeline来达到SOTA performance。

为什么要有方法设计的系统性思路

- 一个清楚的方法设计思路能更有效地解决问题并达到SOTA。
- 让我们提出的方法更有动机，同时使得论文的故事更吸引人。
- 保证了方法的技术创新性，以免审稿人说它没有novelty。
- 同时，该系统性思路可以有效提升我们的技术insights。

一个理想的方法

work程度



技术创新程度

技术创新强的方法
为什么更好？

因为它通常有较大的创新空间。

在此基础上，follower可以比较容易地作出改进，并发表论文。

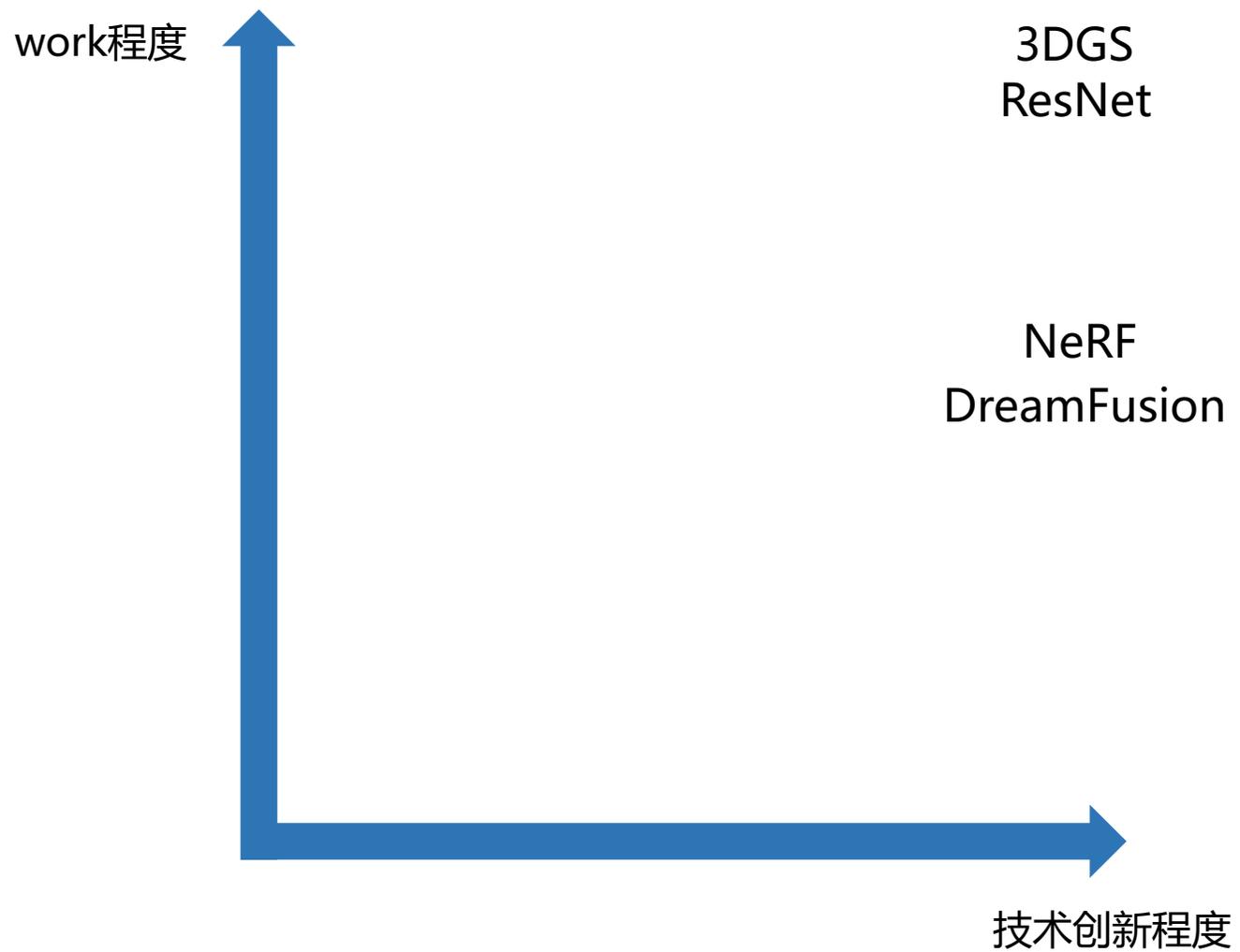
因此有更大的影响力。

怎么判断技术创新的程度

五个程度的技术创新：

1. 一个崭新的范式（新的学习模型，新的表示）
2. 基于已有pipeline提出非常新颖的模块
3. 将几个已有的模块组合成新的pipeline
4. 在已有pipeline中加入经过简单改进的模块
5. 在已有pipeline中加入现有的模块

案例



如何设计方法

分析原因

- 遵从第一性原理，分析为什么当前的SOTA效果不够好。

设计方法

- 根据分析的原因，设计一种能达到SOTA的方法。

判断合理性

- 判断提出的技术方法是否合理

改进方法

- 通过读论文、讨论和做实验改进技术方法

分析原因

分析原因

- 遵从第一性原理，分析为什么当前的SOTA效果不够好。

设计方法

- 根据分析的原因，设计一种能达到SOTA的方法。

判断合理性

- 判断提出的技术方法是否合理

改进方法

- 通过读论文、讨论和做实验改进技术方法

我们需要寻找什么原因，以及为什么？

目标原因

- 为什么目前的SOTA在某某方面效果不好？

为什么我们需要寻找这样的原因

- 第一性原理：找到本质原因，对症下药。
- 有目的地设计方法至关重要。

什么是第一性原理：马斯克的例子

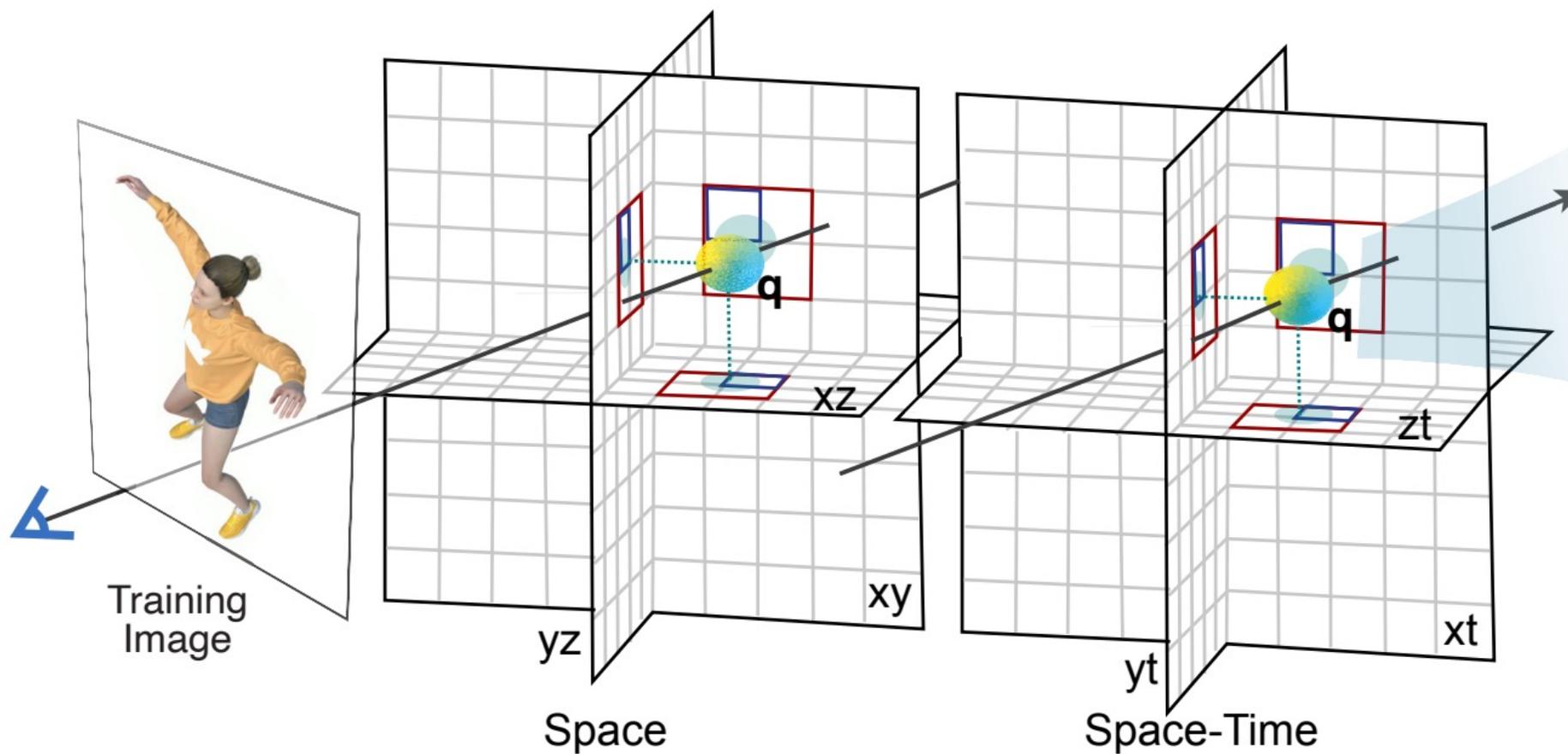
- 特斯拉研制电动汽车期间，曾遇到一个难题：**电池成本居高不下**。当时储能电池的的市场价格是每千瓦时600美元，这个市场价格很稳定，短期内不会有太大的变动。
- 但是马斯克从**第一性原理**角度进行思考：**电池成本为什么这么贵，电池组到底是由什么材料组成的？**这些电池原料的市场价格是多少？如果我们购买这些原材料然后组合成电池，需要多少钱？这个答案是，每千瓦时只需要80美元。
- 从最本质出发，马斯克让电动车的商业化成为可能。

案例：4K4D – 论文任务



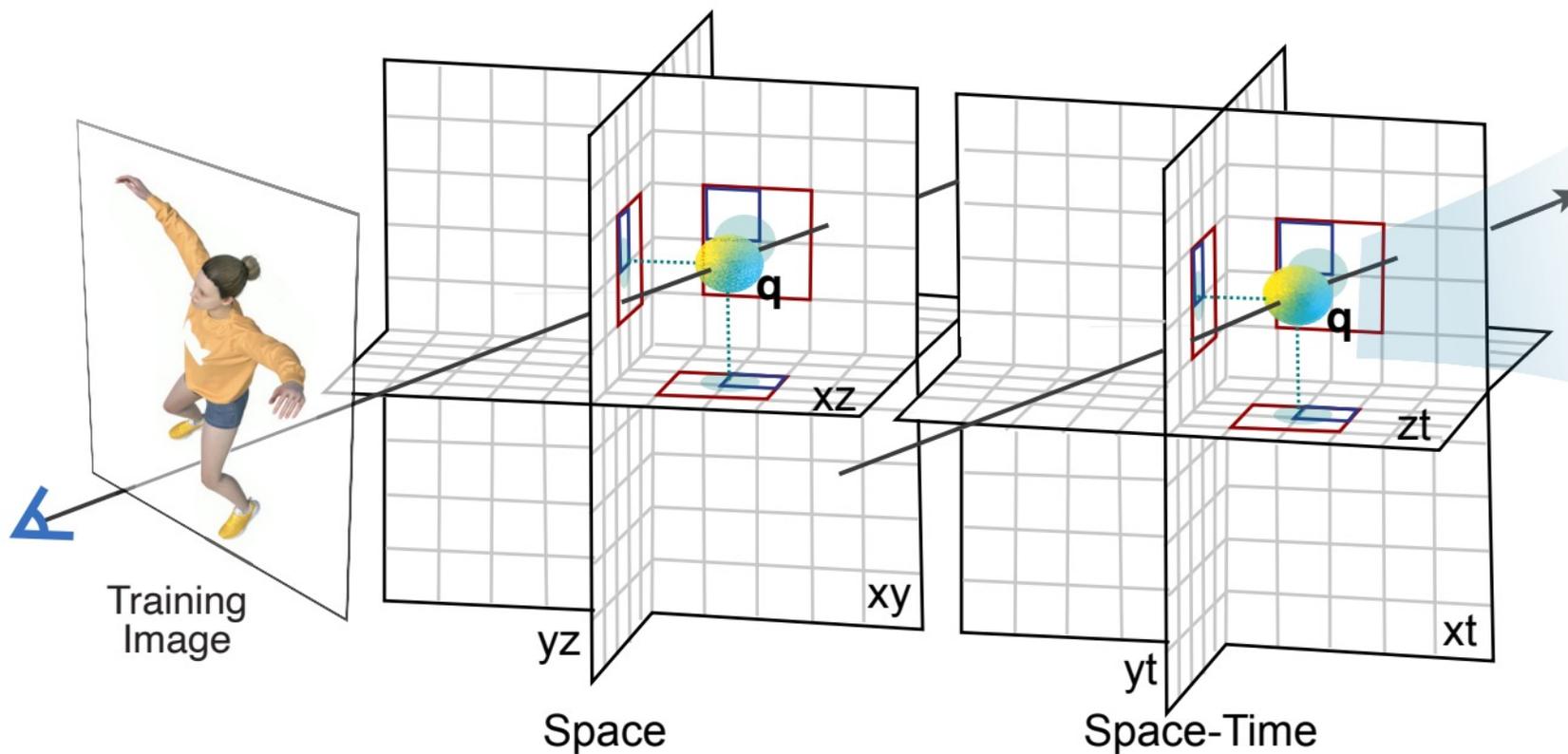
案例：4K4D

- 已有SOTA：K-Planes，遇到的问题速度是慢



案例：4K4D

- 已有SOTA：K-Planes，遇到的问题速度是慢
- 如何改进K-Planes -> 遵从第一性原理，分析其速度慢的原因
 - a) 采样点多; b) 逐点渲染速度慢。



如何找到技术原因

深入学习 SOTA算法

1. 跑代码，学习算法细节。
2. 收集算法的 failure case，整理 failure case 的 pattern，分析其中规律。

做实验获取 更多观测

- 做消融实验，定位哪些模块或者哪些数据导致了问题。

科研经验 带来的直觉

- 直观的原因是什么？

持续思考

- 根据实验现象，不断思考根本的技术原因。

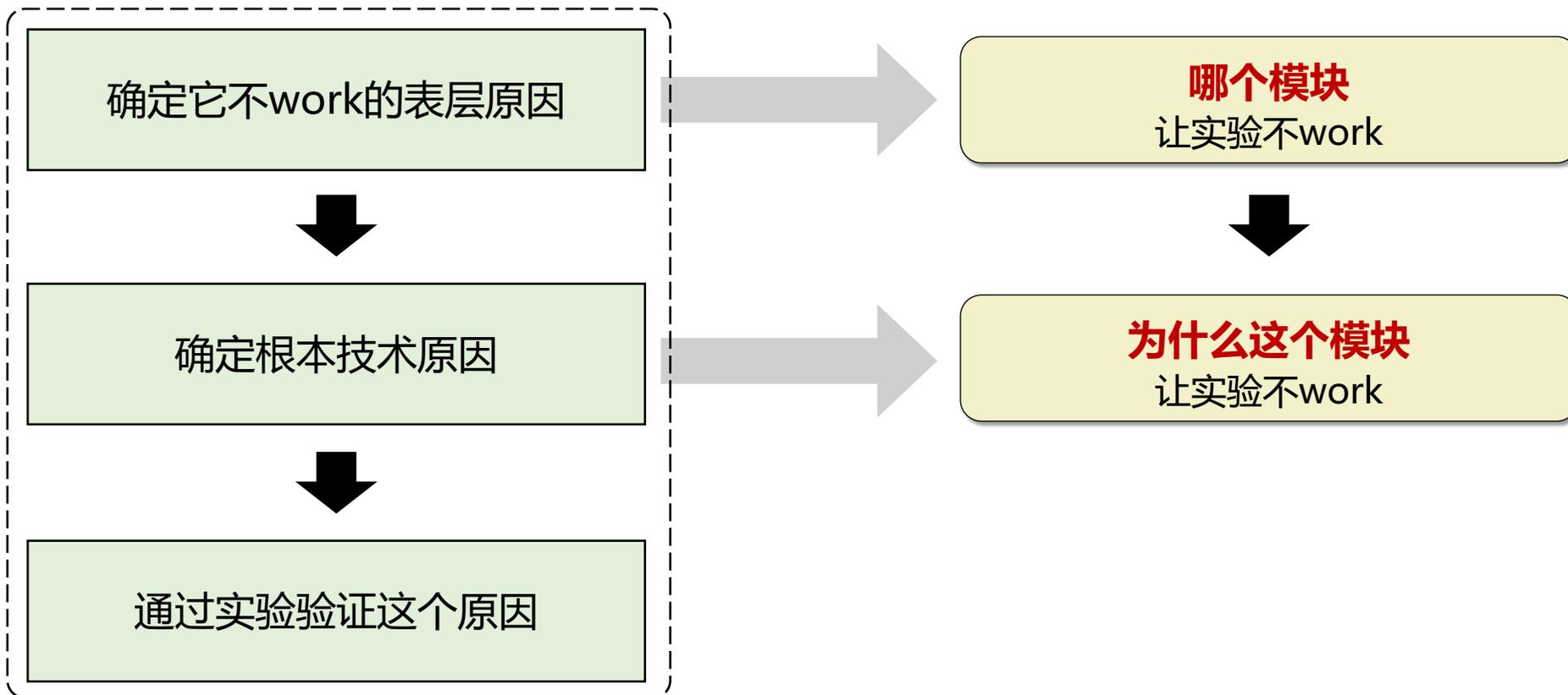
文献调研

- 阅读相关论文，查看是否有工作分析了这个问题。

讨论

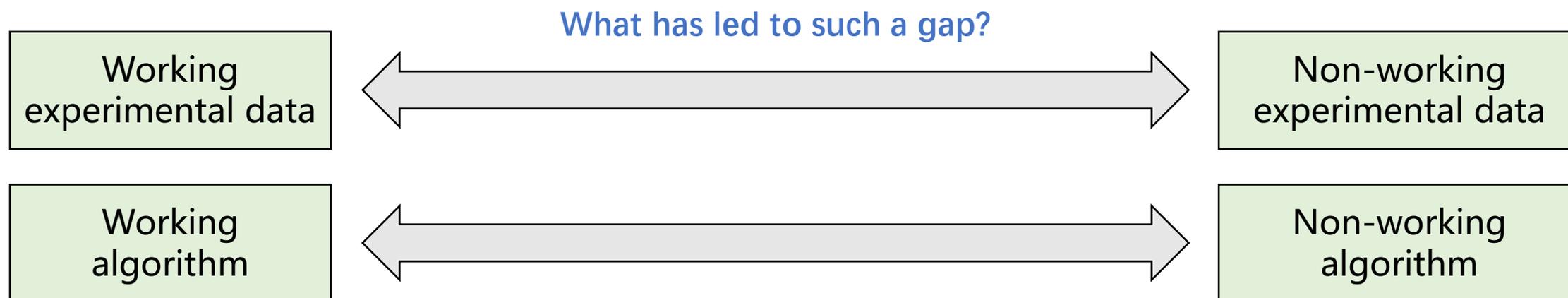
- 经常和他人交流，获得不同的观点和 insights。

如何分析SOTA不work的原因

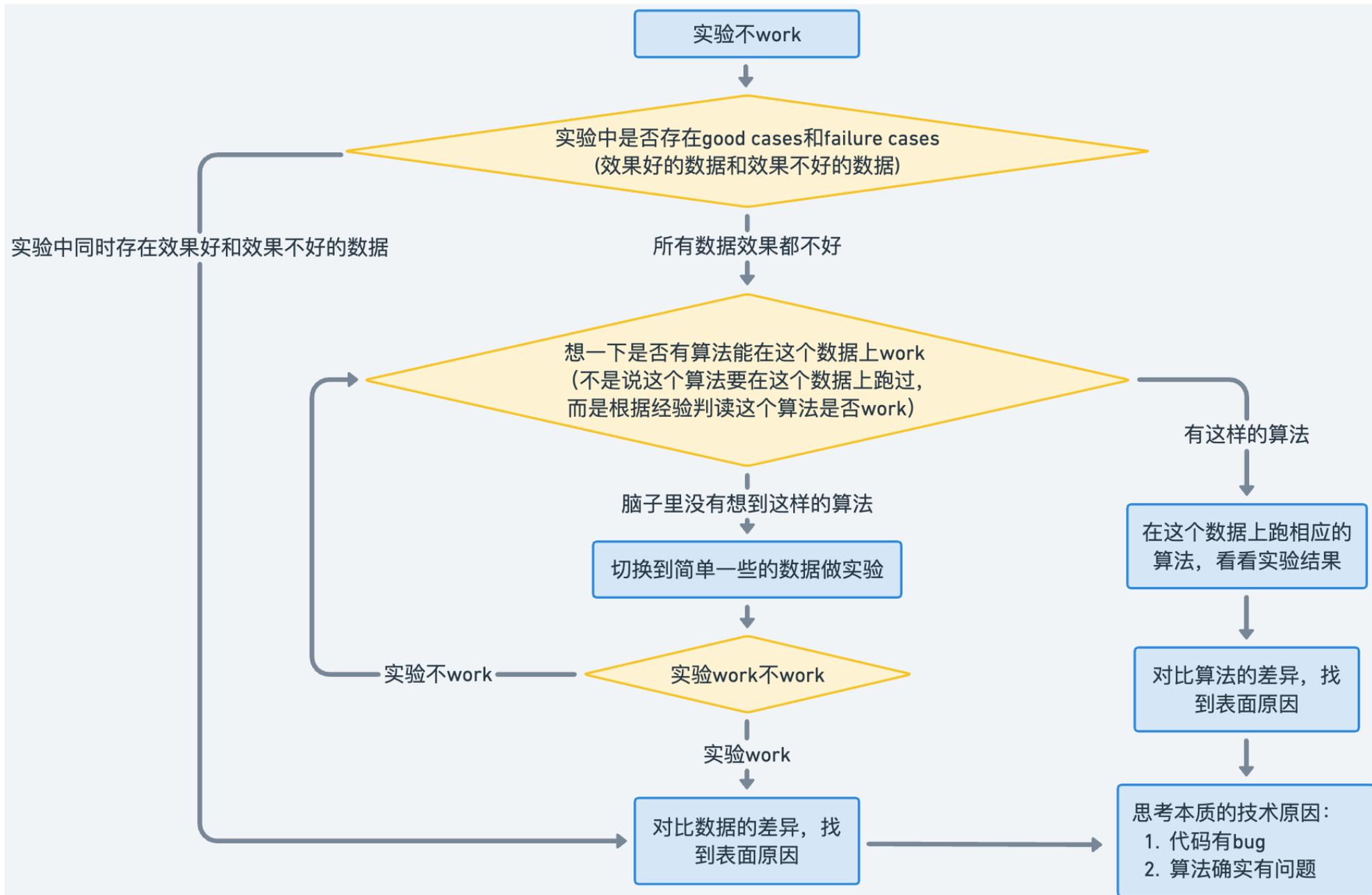


如何发现表层原因：二分查找方法

- 首先确认那些能够work的数据或算法, 然后将其与不work的数据或算法进行比较, 查看他们之间的差异。



如何发现表层原因：二分查找方法



如何发现根本技术原因

确定它不work的表层原因



确定根本技术原因



通过实验验证这个原因

为什么这个模块
让实验不work

为什么切换到某个数据集后
性能会下降？

为什么添加了某个模块后
性能会下降？

为什么改变了某个参数后
性能会下降？

尽可能地列出可能的技术原因



如何发现根本技术原因

SOTA不work通常有两个技术原因：

1. 我们修改的代码中有bug。
2. “xx 算法” 应用在 “xx 数据” 确实存在问题。

如何调试代码

步骤1：彻底理解算法/代码（调试的基础）



步骤2：确定能够持续重现bug的输入和设置



步骤3：基于二分查找法缩小bug的范围

TURING 图灵程序设计丛书

AMACOM

Debugging

The 9 Indispensable Rules for Finding

Even the Most Elusive Software and Hardware Problems

调试九法

软硬件错误的排查之道

[美] David J. Agans 著
赵俐 译



人民邮电出版社
POSTS & TELECOM PRESS

如何调试代码

在将bug锁定在适当的范围内后：

步骤4：逐行调试

- 观察操作的细节。
- 使用控制变量法最终确定bug。

TURING 图灵程序设计丛书

AMACOM

Debugging

The 9 Indispensable Rules for Finding

Even the Most Elusive Software and Hardware Problems

调试九法 软硬件错误的排查之道

[美] David J. Agans 著
赵俐 译

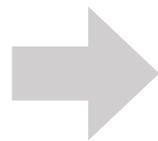


人民邮电出版社
POSTS & TELECOM PRESS

如何调试算法

步骤1：查明导致问题的模块

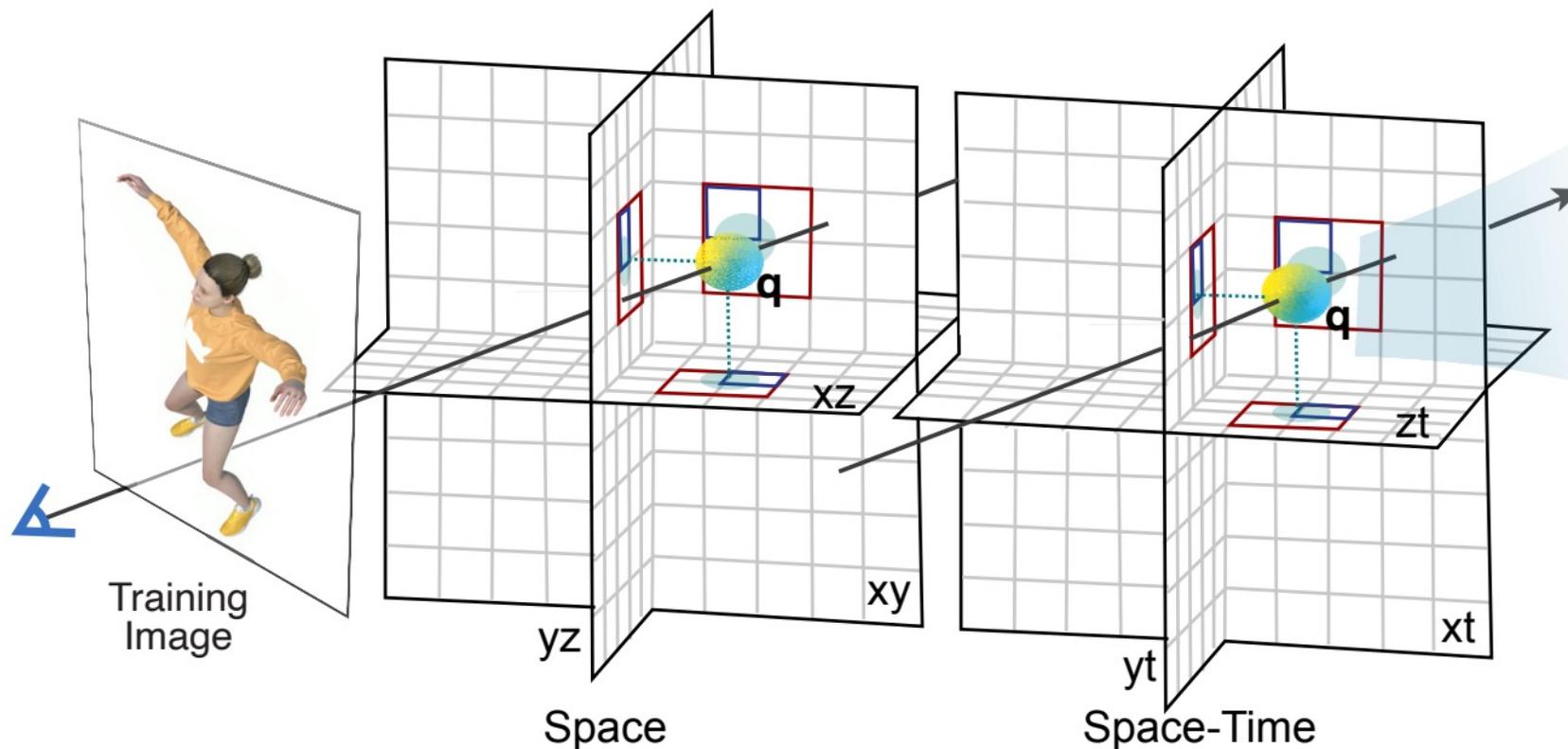
根据二元搜索方法缩小
可能存在问题的模块范围



根据变量控制方法确定
导致问题的特定模块（消融实验）

案例：4K4D

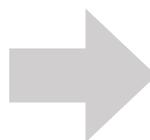
- 4K4D Project中怎么定位K-Planes速度慢的原因
 - K-Planes包含的模块：Ray Sampling、Point Prediction、Volume Rendering



如何调试算法

步骤1：查明导致问题的模块

根据二元搜索方法缩小
可能存在问题的模块范围



根据变量控制方法确定
导致问题的特定模块（消融实验）



步骤2：分析根本技术原因

依靠你积累的技术技能和经验
来推断原因

和他人讨论推断原因

如何与他人讨论实验结果

有效且高效讨论的五个要点

实验目的

描述为什么要做这个实验，同时你希望通过这个实验获得什么。

实验设置

实验中用了什么样的数据，同时对算法做出了哪些改变。

实验现象

列出有明显规律的实验现象。

能确定的
当前问题的范围

猜测的
可能导致的原因

询问是否有其他想法

分析实验结果的典型问题列表

代码调试检查表

各模块的中间输出结果

每个模块代码的中间输出结果是否满足预期？

代码bug的典型原因

积累一些导致代码bug的典型原因（如NaN值，段错误等）。

分析实验结果的典型问题列表

算法调试检查表（与数据相关的问题）

在简单数据上的表现

1. 在training data上的表现如何？
2. 在 xx data上的表现如何 (simpler one)?

两个数据集之间的差异

我们在 xx data有更好的结果。
为什么在这个数据上不能work？
这两个数据集之间有什么差异？

分析实验结果的典型问题列表

算法调试检查表（算法相关问题）

找到一个能够work的算法

我记得xx算法在这个数据集上的表现并不差。为什么我们的方法不work？这需要进行消融实验来检查哪个模块有问题。

检查各模块输出是否正确

我们是否可视化了xx模块的输出？检查输出结果是否满足预期或者是否有任何奇怪的现象发生。

分析实验结果的典型问题列表

算法调试检查表



观察failure cases中是否存在特定的pattern ?



使用3D可视化工具进行调试 (如Viser、Wis3D)

设计方法

分析原因

- 遵从第一性原理，分析为什么当前的SOTA效果不够好。

设计方法

- 根据分析的原因，设计一种能达到SOTA的方法。

判断合理性

- 判断提出的技术方法是否合理

改进方法

- 通过读论文、讨论和做实验改进技术方法

设计方法

为什么需要一个系统性的方法论

- 为了使创新和设计解决方案的过程更有迹可循。

一个例子

- 研究发现，在稀疏视角下，由于观测不足，NeRF的表现很差，导致优化陷入了局部最小值。**怎样才能设计出一种有效且创新的方法呢？**

设计方法

一个事实

- 我们不得不承认，所有的方法都是在创新地组合已有的技术。

案例分析

- ResNet = CNN + Skip Connection
- NeRF = Neural Field + Radiance Field
- DreamFusion = ClipNeRF + SDS Loss
- **你能提供一个反例吗?**

如何设计方法

列出原因

列出本质技术原因：为什么原有SOTA方法不work？

列出技术

列出所有与技术原因相关的、有可能能解决该问题的技术

选择新技术

选择近年来的新技术，需要是未被well-explored、有想象空间的技术

不断积累并等待灵感爆发

等待灵感爆发的时刻，脑海中涌现出一个有效且有意思的pipeline

讨论

与周围的人进行广泛的讨论，获得insights，迭代优化pipelines

设计方法

每天仔细思考，如何基于这些技术构造pipeline？



案例：4K4D

- 已有SOTA：K-Planes，遇到的问题速度是速度慢
- 如何改进K-Planes -> 遵从第一性原理，分析其速度慢的原因
 - a) 采样点多; b) 逐点渲染速度慢。
- 相关的技术：Voxel-based Rendering、Point Rendering、Depth Guidance

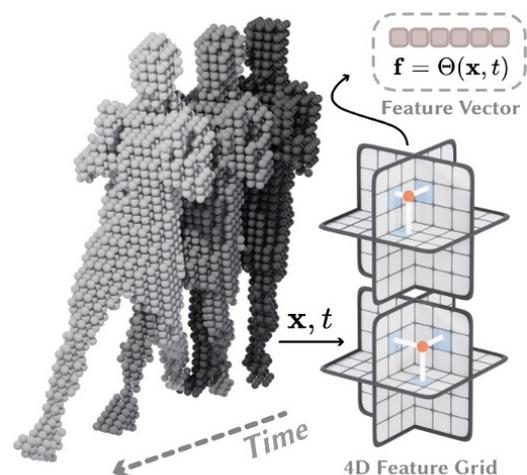


案例：4K4D

- 已有SOTA：K-Planes，遇到的问题速度是速度慢
- 如何改进K-Planes -> 遵从第一性原理，分析其速度慢的原因
 - a) 采样点多; b) 逐点渲染速度慢。
- 相关的技术：Voxel-based Rendering、Point Rendering、Depth Guidance
- 选择新技术：发现在动态场景，Point Rendering还未被较多探索

案例：4K4D

- 已有SOTA：K-Planes，遇到的问题速度是速度慢
- 如何改进K-Planes -> 遵从第一性原理，分析其速度慢的原因
 - a) 采样点多; b) 逐点渲染速度慢。
- 相关的技术：Voxel-based Rendering、Point Rendering、Depth Guidance
- 选择新技术：发现在动态场景，Point Rendering还未被较多探索
- 设计方法：Point Cloud Sequence



(a) Point Cloud Sequence

如何设计一个好方法

一个事实

- 好的方法通常由**某些新技术**组成。

案例分析

- ResNet = **CNN** + Skip Connection
- NeRF = **Neural Field** + Radiance Field
- DreamFusion = ClipNeRF + **SDS Loss**
- 你能给出一个反例吗?

设计方法的三个方面的能力

技术积累

- 广泛的技术积累是方法设计的基础

深入思考

- 对技术和技术挑战的深刻理解

敏锐的洞察力

- 掌握前沿的技术
- 敏锐地发现如何使用它们解决已有问题

判断合理性

分析原因

- 遵从第一性原理，分析为什么当前的SOTA效果不够好。

设计方法

- 根据分析的原因，设计一种能达到SOTA的方法。

判断合理性

- 判断提出的技术方法是否合理

改进方法

- 通过读论文、讨论和做实验改进技术方法

判断提出的方法的合理性



多与他人沟通来获得反馈



阅读论文获得相关技术的分析

改进方法

分析原因

- 遵从第一性原理，分析为什么当前的SOTA效果不够好。

设计方法

- 根据分析的原因，设计一种能达到SOTA的方法。

判断合理性

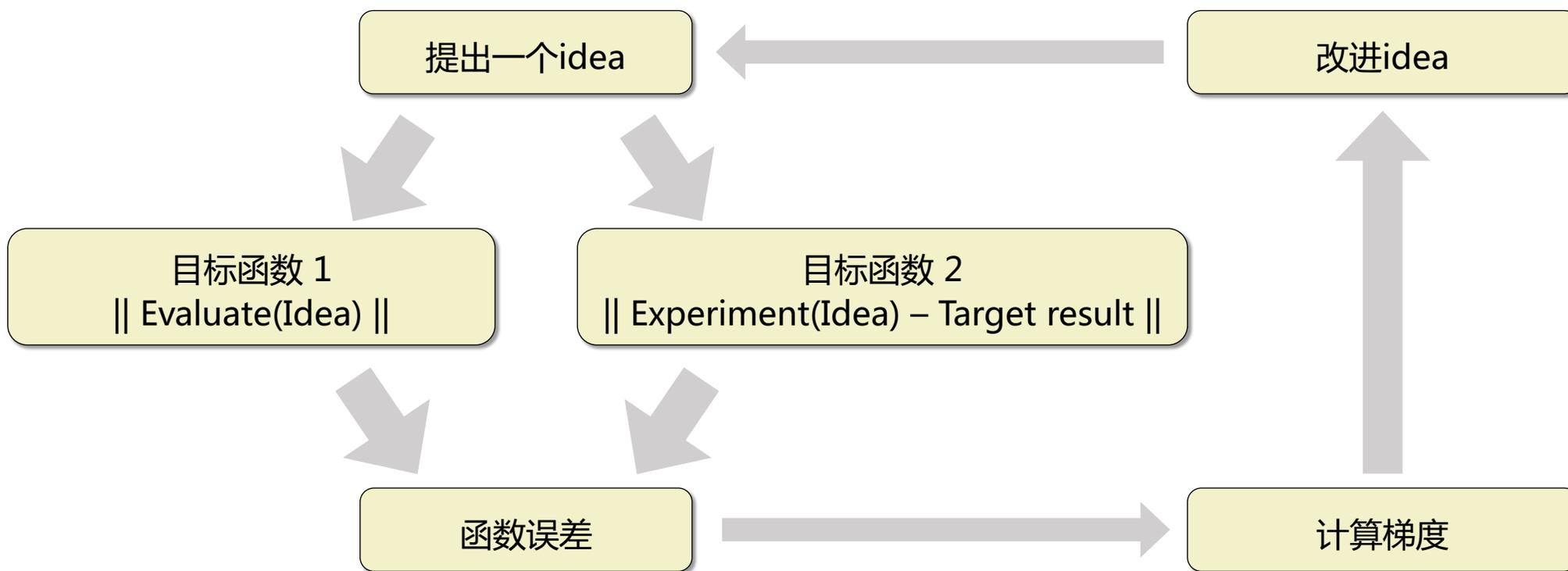
- 判断提出的技术方法是否合理

改进方法

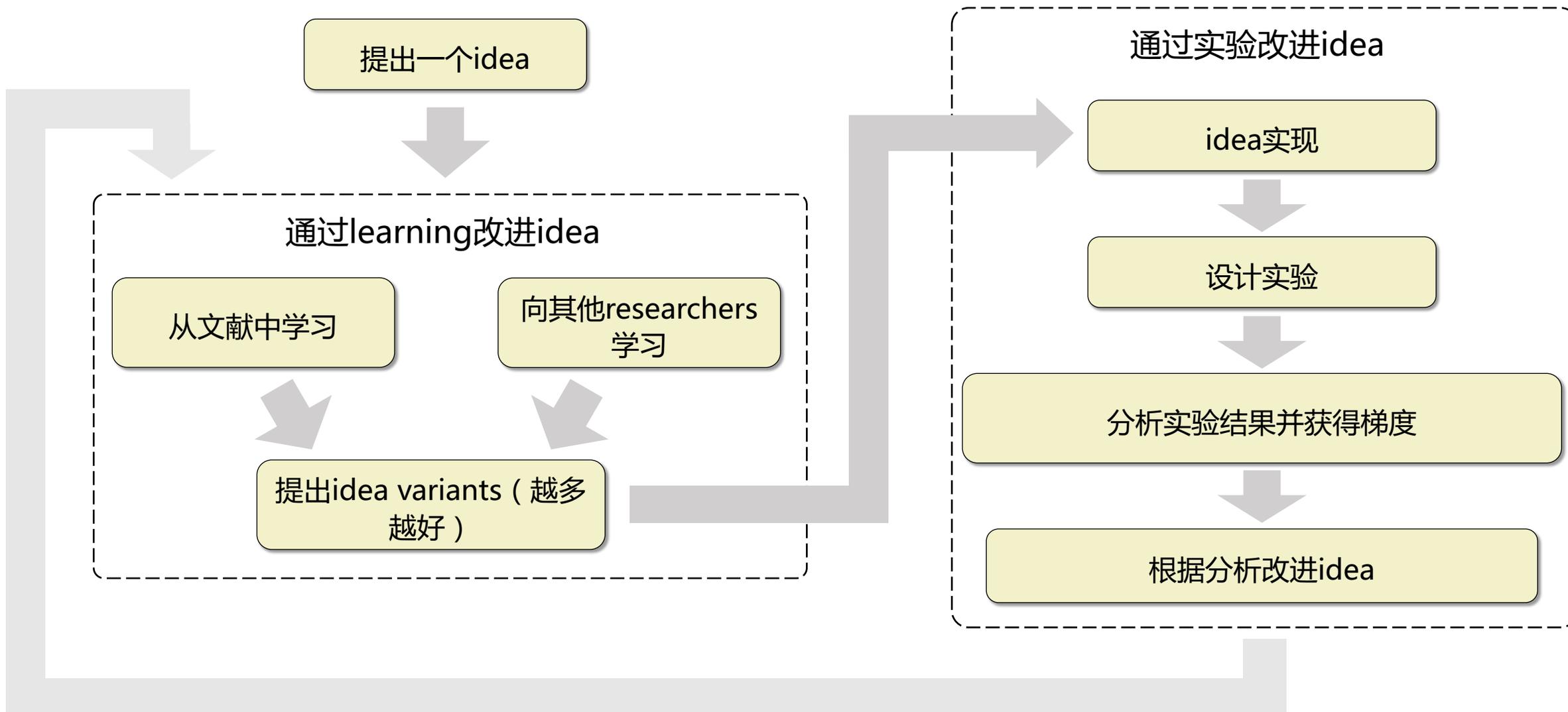
- 通过读论文、讨论和做实验改进技术方法

如何通过实验改进方法

- 可以把这个过程当做是随机梯度下降

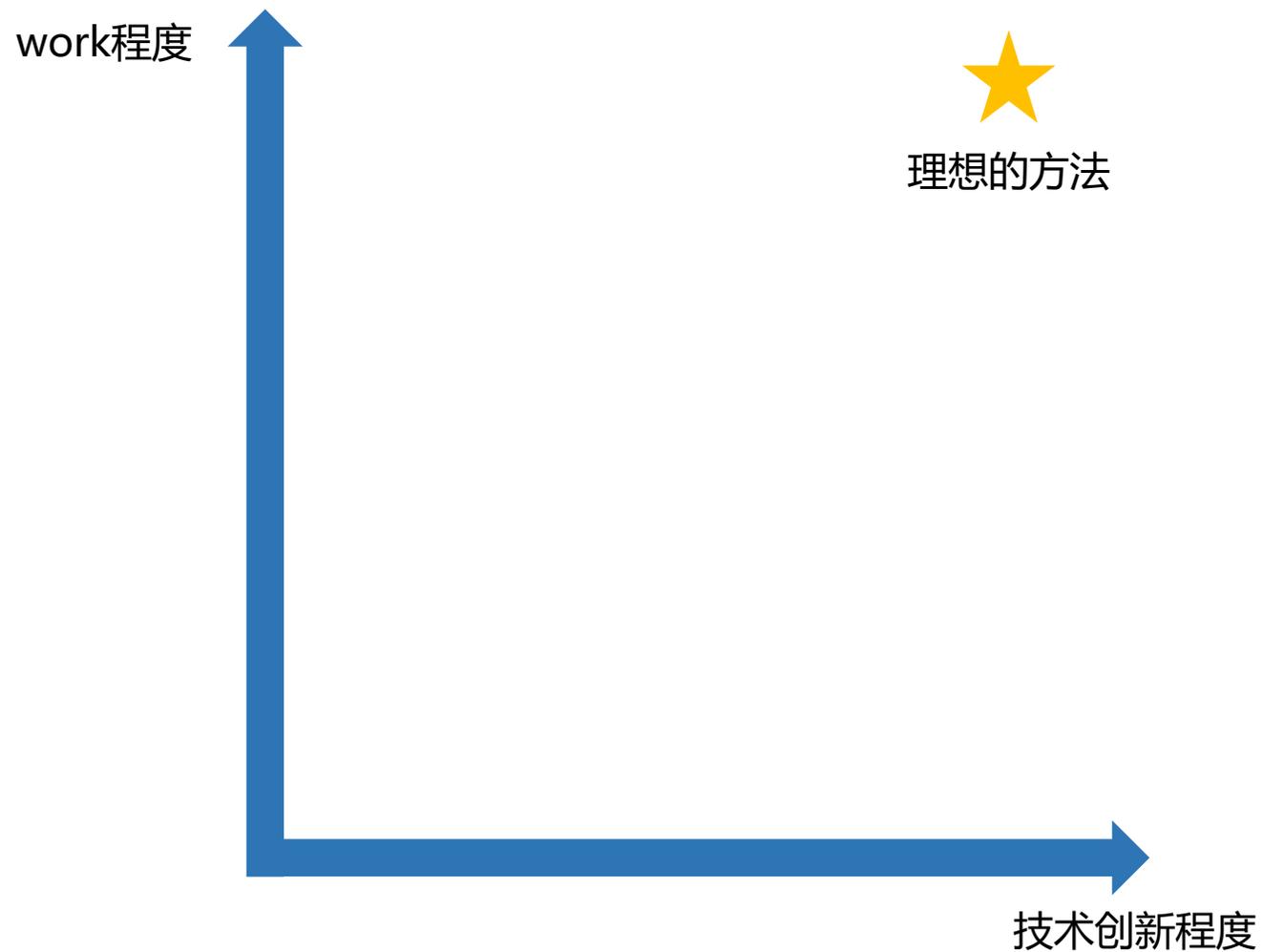


如何通过实验改进方法



设计方法时的其他要点

理想的方法：用新技术解决问题



设计方法时的其他要点

什么是理想的方法

- 用新技术解决问题

背后的逻辑

- “新技术”和“有效性”一般来说有很强的关联
- 改进现有技术通常不会带来性能上的重大突破（不然已经被其他研究人员发现了）
- 对于一个想解决有趣科学问题的前沿项目，如果不做方法创新，只停留在原来的技术水平上，就很可能无法带来明显的提升。

设计方法时的其他要点

五个程度的技术创新：

1. 一个崭新的范式（新的学习模型，新的表示）
2. 基于已有pipeline提出非常新颖的模块
3. 将几个已有的模块组合成新的pipeline
4. 在已有pipeline中加入经过简单改进的模块
5. 在已有pipeline中加入现有的模块

设计方法时的其他要点

不同阶段的人应该追求什么样的方法

- 任意阶段的研究人员（初学者、大佬）都可以把使用新技术解决问题作为目标
- 只要想法多，初学者也能设计出创新且有效的pipeline

一个简单高效的设计方法的方式

- 利用新技术解决自己领域的技术难题

新技术

应用

你的领域

一个简单高效的设计方法的方式

为什么这个方式好

- 技术新颖：论文很可能被录用。
- 简单直接：就算是低年级学生也能做。
- 启发性强：发掘新技术的应用潜力。
- 先进前沿：如果新技术能work，就会有很多follow up。

这个方式存在的问题

- 要求研究人员对“技术”有深入的思考，而不是简单拼接、当作工具使用。
- 为了让新技术在一个新领域work，需要深入的思考和改进，因此研究人员要有很强的技术能力。

如何跟进新技术

读论文

- 阅读有关新技术的论文，最好也要跟进机器学习领域的新技术。
- 这类论文很少

看推特

- 看看大家在推特上聊的新论文或新技术

做笔记

- 及时跟进并整理新技术，以免忘记

如何跟进新技术

什么样的技术才算新

- 全新的范式：新的学习模型，新的表示
- 新颖的算法模块。

如何把新技术用到你的领域

分析新技术能解决的问题

- 列出新技术可能可以解决的技术难题。

整理自己领域的技术难题

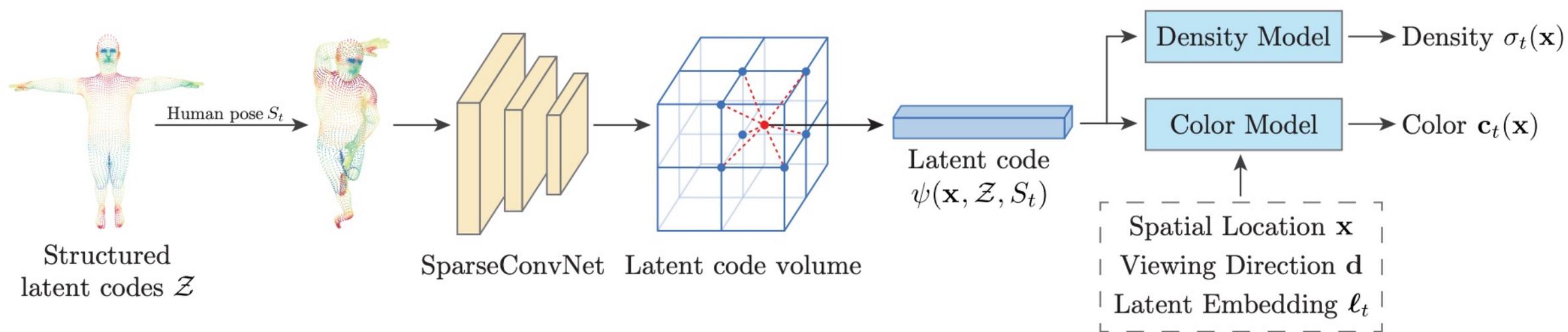
- 列出领域中当前重要的问题，以及整理它们背后本质的技术难题。

关联

- 将新技术和当前领域的技术难题关联起来

案例：Neural Body

- 第一个将NeRF应用到人体建模的技术。
- 技术做法挺直接的，但因为NeRF本身技术潜力大，所以后期改进的人较多。



为什么提倡新技术

学生的视角

- Ph.D学生的目标:
 - 努力成为一个专业能力很强的人，对新技术具有出色的学习、思考与分析能力。
 - 能够深入挖掘new and under-explored的技术，改进该技术以解决自己领域前沿问题。

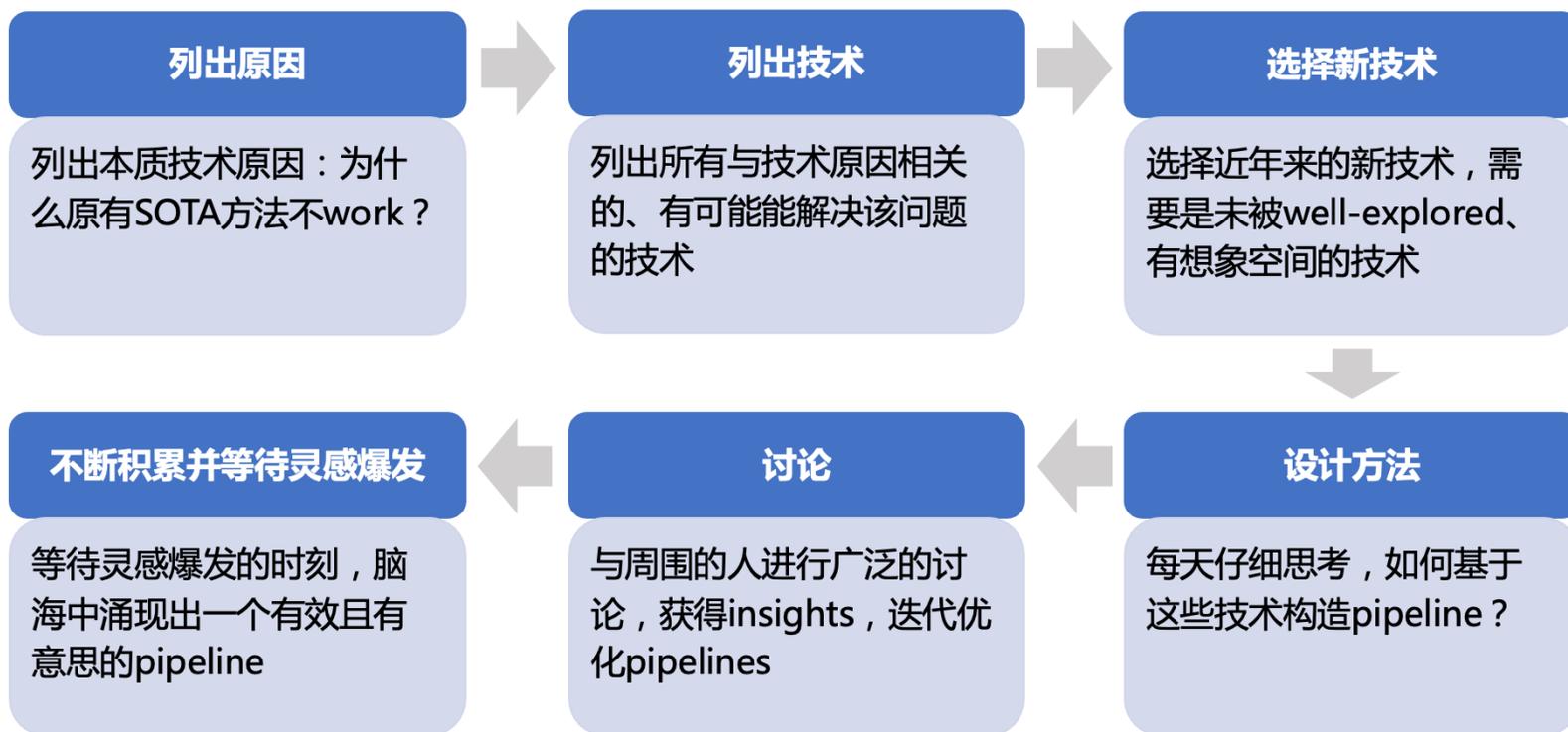
为什么提倡新技术

审稿人的视角

- 在我看来，审稿人希望论文有技术贡献，可能是因为他们想看到更多的技术创新和分析。
- 如果一篇论文只是讲故事，但是不介绍新技术，它对这个领域其实没作出技术贡献，对于领域而言，没有技术上的增量。

作业

- 针对自己在研究的课题，设计一个初步的方法，并给出方法设计的具体思路。
- 提交格式：PDF文件。





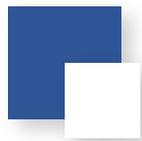
QA：前置基础

1. 在科研初期往往总会要求打好数理基础很重要，如何打好数理基础？
2. 如何提升代码能力？
3. 建立领域视野时，非常容易陷入某篇论文的工作，导致需要很久才能建立起领域视野怎么办？
4. 是否需要知晓SOTA相比于其他方法的改进在何处？
5. 看过的做过笔记的论文还是会忘记，有时候会怀疑大量看论文的意义在哪



QA：如何分析SOTA

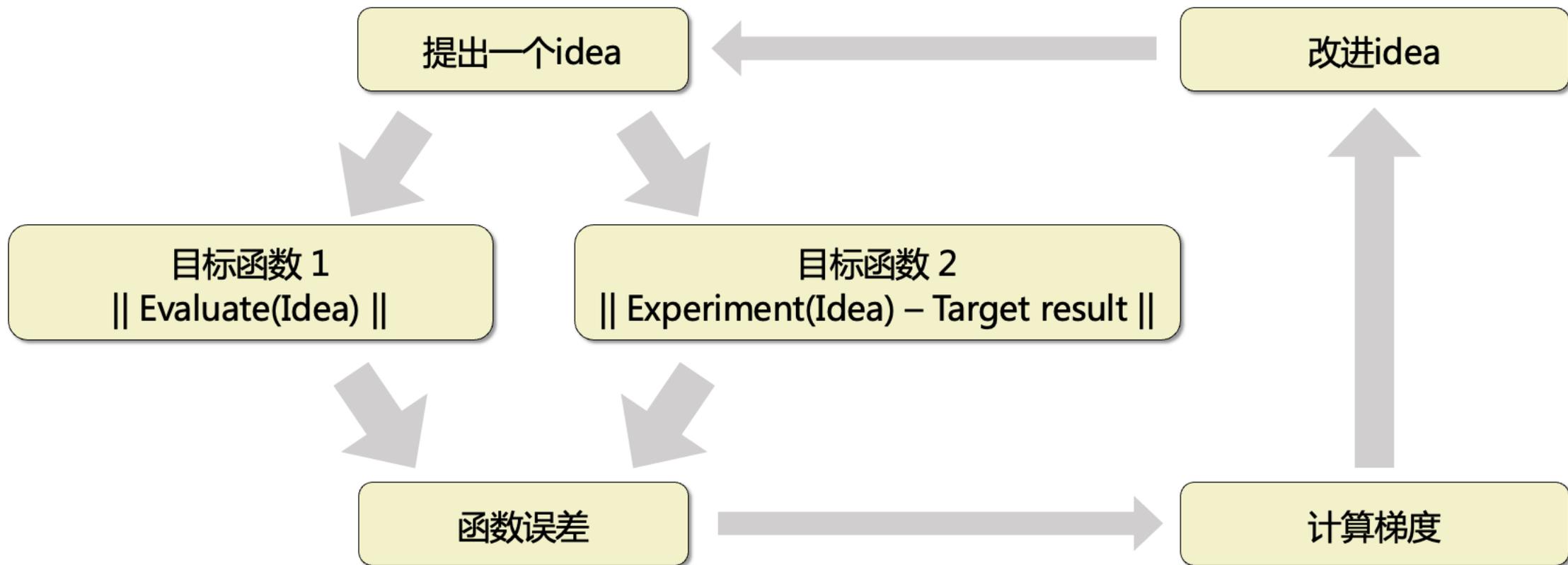
1. 如何识别论文方法的不足？
2. 如何挑选baseline？
3. 如果领域主要针对场景进行创新，如何确定领域的SOTA？
4. 如何把握第一性原则？如何判断自己观念中的第一性原则是否正确？
5. 如何知道当前的方法是不是足够好了，有没有进一步挖掘的价值？
6. 是否需要足够的理论知识储备和广阔的视野 + 对该论文的技术复现十分深刻才可以去做缺点不足的分析？



QA：如何跟进新技术

1. 如何判断什么是"新"技术？
2. 跟进新技术是一个完全不熟悉的领域，如何快速理解新技术呢？

QA：如何迭代方法





QA：方法Novelty

1. 强调基于SOTA的改进是否意味着这种思路下的研究只是针对已有任务的性能提升，可能缺乏创新性？
2. 确定要解决的核心问题后，如何设计新颖的技术方法？通过阅读相关领域的论文总感觉跳不出去，缺乏创新。



QA：实验设计

1. 技术方案要到什么粒度才开始实验步骤？
2. 如何进行实验设计？



QA：和人讨论

1. 和他人交流时，往往觉得自己还没办法很明确地说出自己的问题。平时是不是需要常常总结问题、提炼问题？
2. 和学长学姐沟通时，怎么能够更有效地交流？
3. 和他人交流时，是不是都要准备得比较充分了再去交流比较好，还是多呈现自己的问题？



QA：面向本科生的问题

1. 本科生参与科研是否存在知识储备上的局限性？
2. 这门课程可以传达什么最有帮助的观念给感兴趣的本科生呢？
3. 对以后想申请3D方向PhD的本科生，此时更需要去注重什么基础呢？



谢谢!



彭思达



高俊



彭崧猷



王倩倩